ICASE

STABLE IMPLICIT FINITE-DIFFERENCE METHODS FOR

THREE-DIMENSIONAL HYPERBOLIC SYSTEMS

Saul S. Abarbanel

Douglas L. Dwoyer

David Gottlieb

Report No. 82-39

November 26, 1982

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING

NASA Langley Research Center, Hampton, Virginia 23665

Operated by the

UNIVERSITIES SPACE USRA RESEARCH ASSOCIATION

STABLE IMPLICIT FINITE-DIFFERENCE METHODS FOR THREE-DIMENSIONAL HYPERBOLIC SYSTEMS

Saul S. Abarbanel
Tel-Aviv University

Douglas L. Dwoyer
NASA Langley Research Center

David Gottlieb
Institute for Computer Applications in Science and Engineering
and
Tel-Aviv University

ABSTRACT

The numerical stability of conventional implicit algorithms of the approximate factorization type as applied to two-dimensional and three-dimensional hyperbolic systems is analyzed. The unconditional instability of these methods for the three-dimensional wave equation and Euler's equations of gas dynamics is proven and a modified scheme is proposed which is unconditionally stable for the scalar wave equation. Analytical results are verified by numerical experimentation.

Research for the first author was supported in part by the U. S. Air Force Office of Scientific Research under Grant AFOSR-80-0249 and in part by the NASA Cooperative Agreement NCCI-45. Research for the third author was supported in part by the U. S. Army Research and Standardization Group (Europe) under Contract DAJA38-80-C-0032 and in part by the National Aeronautics and Space Administration under NASA Contract Nos. NAS1-16394 and NAS1-15810 while he was in residence at the ICASE, NASA Langley Research Center, Hampton, VA 23665.

1. INTRODUCTION

In recent years there has been an increased interest in numerical solution of the time-dependent Euler equations in two and three spatial dimensions. Because of stability restrictions associated with explicit finite-difference methods, renewed attention has been paid to implicit methods. These methods, in their primitive form, require a prohibitive amount of computational effort per time-step. This difficulty is circumvented by resorting to approximate factorizations such as the Beam-Warming scheme [2]. Beam and Warming employed what is known as the delta-form. This is done in order to avoid, as one marches to steady-state, the time-step dependent splitting errors. Their approach produces a scheme which is analogous in structure to the one proposed by Douglas and Gunn for the parabolic case [3]. In the two-dimensional case, their scheme has been proven to be efficient and second-order accurate in space and (linearly) stable. Unfortunately, this scheme is unstable in three-dimensions for the (scalar) linear wave equation, as shown by Dwoyer and Thames [4] and Warming and Beam [5]. This is also true for the full three-dimensional system of the Euler equations as will be demonstrated later.

The present effort was directed towards developing an implicit scheme for hyperbolic equations which will be stable in three dimensions and yet be free of splitting-errors. A natural starting point was the LODQ method presented in [4] for removal of the splitting-error inherent in locally one-dimensional algorithms (LOD) thereby presenting and alternative to the delta-form of Ref. 2. However, as shown in [4], the LODQ when applied to the three-dimensional wave equation was also unstable, just like the Beam and Warming scheme. This result is not surprising since, as was shown in [4], the two alternatives discussed above (Beam-Warming and LODQ) have the same amplification matrix.

In Section 2, we show how the unstable three-dimensional LODQ can be stabilized without changing or adding to the implicit part of the algorithm. We also show how the delta-form Beam-Warming algorithm can be suitably modified. In Section 3 we discuss programming aspects of the new algorithm while in Section 4 numerical evidence is presented, for the three-dimensional linear wave equation case, that the method is stable and retains its second-order accuracy. While the present synthesis of the scheme is immediately applicable to a nonlinear hyperbolic system its analysis and confirming computations were carried out for the linear scalar case. Work is presently being carried out leading to application to the full Euler system of fluid dynamics. The results will be presented in a subsequent report.

2. THE ALGORITHM AND ITS STABILITY

Consider first the two-dimensional hyperbolic set

$$\frac{\partial \vec{U}}{\partial t} = \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = A \frac{\partial \vec{U}}{\partial x} + B \frac{\partial \vec{U}}{\partial y} , \tag{2.1}$$

where $\vec{U}$, $F(\vec{U})$ and $G(\vec{U})$ are m-component vectors; A and B are, respectively, the Jacobians $\partial \vec{F}/\partial \vec{U}$ and $\partial \vec{G}/\partial \vec{U}$, assumed to be simultaneously symmetrizable. This assumption is valid, e.g., for Euler's equations. The standard, approximate-factored, form of the Backward Euler algorithm applied to (2.1) takes the form

$$(I - \Delta t A \delta_x \mu_x)(I - \Delta t B \delta_y \mu_y) U^{n+1}_{j,k} = U^n_{j,k}, \tag{2.2}$$

where $U^n_{j,k} = U(j\Delta x, k\Delta y, n\Delta t)$ is the solution vector described over a

rectangular mesh, and δ_x and μ_x are respectively a differencing and averaging operators given by $\delta_x U^n_{j,k} = (U^n_{j+1/2,k} - U^n_{j-1/2,k})/\Delta x$, $\mu_x U^n_{j,k} = (U^n_{j+1/2,k} + U^n_{j-1/2,k})/2$. Analogous definitions hold for δ_y and μ_y.

That (2.2) will, at steady-state, yield a result which is Δt-dependent is easily seen by writing it as

$$\frac{U^{n+1}_{j,k} - U^n_{j,k}}{\Delta t} = (A\delta_x\mu_x + B\delta_y\mu_y)U^{n+1}_{j,k} - \Delta t AB\delta_x\mu_x\delta_y\mu_y U^{n+1}_{j,k} \ . \qquad (2.3)$$

The last term on the right-hand side is the splitting error, and it becomes unacceptable as larger Δt's are utilized in the time-advance. In three dimensions the splitting error is proportional to Δt^2.

In order to remove the splitting error, one may use any of several strategies - we will briefly review two of them. In the delta-form, [2], one uses instead of (2.2)

$$(I-\alpha\Delta t A\delta_x\mu_x)(I-\alpha\Delta t B\delta_y\mu_y)(U^{n+1}_{j,k}-U^n_{j,k}) = \Delta t(A\delta_x\mu_x + B\delta_y\mu_y)U^n, \qquad (2.4)$$

where α is a free parameter (usually $1/2 \leq \alpha \leq 1$).

If we achieve convergence, then indeed the right-hand side of (2.4) solves (to second-order in the spatial coordinates) the steady-state operator. Another approach (LODQ) was presented in Ref. [4]. The "given" data at any time level is designated $Q^n_{j,k}$. One then first advanced temporally by using the standard Backward Euler

$$(1-\alpha\Delta t A\delta_x\mu_x)(1-\alpha\Delta t B\delta_y\mu_y)U^{n+1}_{j,k} = Q^n_{j,k} \ , \qquad (2.5a)$$

and follows it by an explicit step

$$Q^{n+1}_{j,k} = Q^{n}_{j,k} + \Delta t\left(A\delta_x\mu_x + B\delta_y\mu_y\right)U^{n+1}_{j,k} \ . \tag{2.5b}$$

Clearly, if steady state is achieved then $U^{n+1}_{j,k}$ and not $Q^{n+1}_{j,k}$ is the (second-order accurate) solution to the steady-state operator.

In the scalar case (2.4) and (2.5) are equivalent, as an examination of their respective amplification factors will show. For this case it is also easy to establish stability, see [4].

In the three-dimensional case, the situation is different. Both the delta-form of the Backward Euler scheme and the LODQ form are unconditionally unstable for a general scalar case and also, more importantly, in the case of the system of Euler's equations. This will now be demonstrated. The three-dimensional Euler equations for gas-dynamics may be written as follows:

$$\frac{\partial \vec{V}}{\partial t} + A\,\frac{\partial \vec{V}}{\partial x} + B\,\frac{\partial \vec{V}}{\partial y} + C\,\frac{\partial \vec{V}}{\partial z} = 0, \tag{2.6}$$

where V is the column vector (ρ,u,v,w,p) and the coefficient matrices for an ideal gas are given by:

$$A = \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 1/\rho \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & 0 & u \end{bmatrix}, \quad B = \begin{bmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 1/\rho \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \gamma p & 0 & v \end{bmatrix}, \quad C = \begin{bmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & 1/\rho \\ 0 & 0 & 0 & \gamma p & w \end{bmatrix}. \tag{2.7}$$

The various parameters and dependent variables ρ,u,v,w,p and γ are, respectively the density, the x, y, and z components of velocity, the pressure and the ratio of specific heats at constant pressure and volume.

The matrix coefficients, A, B, and C are simultaneously symmetrizable via a similarity transformation [1]. It is given by:

$$S = \begin{bmatrix} \beta\rho & \rho & 0 & 0 & \rho \\ 0 & c & 0 & 0 & -c \\ 0 & 0 & \sqrt{2}c & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2}c & 0 \\ 0 & \rho c^2 & 0 & 0 & \rho c^2 \end{bmatrix}, \quad S^{-1} = \begin{bmatrix} 1/\beta\rho & 0 & 0 & 0 & -1/\beta\rho c^2 \\ 0 & 1/2c & 0 & 0 & 1/2\rho c^2 \\ 0 & 0 & 1/\sqrt{2}c & 0 & 0 \\ 0 & 0 & 0 & 1/\sqrt{2}c & 0 \\ 0 & -1/2c & 0 & 0 & 1/2\rho c^2 \end{bmatrix}, \tag{2.8}$$

where $c = (\gamma p/\rho)^{1/2}$ is the speed of sound and $\beta = \sqrt{2(\gamma-1)}$. The form of the symmetrized coefficients is important for demonstrating the numerical instability of the scheme, and so they are presented explicitly herewith:

$$A_s = S^{-1}AS = \begin{bmatrix} u & 0 & 0 & 0 & 0 \\ 0 & u+c & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & u-c \end{bmatrix}, \quad B_s = S^{-1}BS = \begin{bmatrix} v & 0 & 0 & 0 & 0 \\ 0 & v & c/\sqrt{2} & 0 & 0 \\ 0 & c/\sqrt{2} & v & 0 & c/\sqrt{2} \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & c/\sqrt{2} & 0 & v \end{bmatrix}$$

$$C_s = S^{-1}CS = \begin{bmatrix} w & 0 & 0 & 0 & 0 \\ 0 & w & 0 & c/\sqrt{2} & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & c/\sqrt{2} & 0 & w & c/\sqrt{2} \\ 0 & 0 & 0 & c/\sqrt{2} & w \end{bmatrix}. \tag{2.9}$$

For the purpose of (linear) stability analysis, one may consider the matrix coefficients as locally constant and then (2.6) takes the form

$$\frac{\partial \vec{U}}{\partial t} + A_s \frac{\partial \vec{U}}{\partial x} + B_s \frac{\partial \vec{U}}{\partial y} + C_s \frac{\partial \vec{U}}{\partial z} = 0, \tag{2.10}$$

with $\vec{U} = S^{-1}\vec{V}$. The delta-form of the approximate-factorization Backward Euler scheme corresponding to (2.4) is then

$$\left(I + \alpha\Delta t A_s \delta_x \mu_x\right)\left(I + \alpha\Delta t B_s \delta_y \mu_y\right)\left(I + \alpha\Delta t C_s \delta_z \mu_z\right)\left(\vec{U}^{n+1}_{j,k,l} - \vec{U}^{n}_{j,k,l}\right)$$

$$= -\Delta t\left(A_s \delta_x \mu_x + B_s \delta_y \mu_y + C_s \delta_z \mu_z\right)\vec{U}^{n}_{j,k,l} \,. \qquad (2.11)$$

Symbolically, we may rewrite (2.11) as (omitting subscripts on the U's):

$$L(\vec{U}^{n+1} - \vec{U}^{n}) = R\vec{U}^{n} \,, \qquad (2.12)$$

with obvious definitions of the matrix operators L and R. Clearly, the amplification matrix for the algorithm represented by (2.11) is

$$G_\Delta = I + \hat{L}^{-1}\hat{R} \,, \qquad (2.13)$$

where $\hat{L}$ and $\hat{R}$ are obtained by evaluating the elements of L and R in Fourier space. The LODQ algorithmic discretization of (2.10) corresponding to (2.12) is

$$L\vec{U}^{n+1} = \vec{Q}^{n} \qquad (2.14a)$$

$$\vec{Q}^{n+1} = \vec{Q}^{n} + R\vec{U}^{n+1} \,, \qquad (2.14b)$$

leading to an amplification matrix of the form

$$G_Q = I + \hat{R}\hat{L}^{-1} \,. \qquad (2.15)$$

Note that only in the scalar case (or in the unlikely event that L^{-1} and R commute) is $G_\Delta = G_Q$. Nonetheless, we shall now show that for the system of the gas-dynamic equations both the three-dimensional Beam-Warming algorithm (2.13), and the LODQ one (2.14), are unstable. We start by noting

that A_s, B_s, and C_s have the property that all the elements in their top row and left columns are zero except the element in the upper left corner; i.e.,

$$A_s = \left[\begin{array}{c|cccc} u & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & \tilde{A}_s & & \\ 0 & & & & \\ 0 & & & & \end{array}\right], \quad B_s = \left[\begin{array}{c|cccc} v & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & \tilde{B}_s & & \\ 0 & & & & \\ 0 & & & & \end{array}\right], \quad C_s = \left[\begin{array}{c|cccc} w & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & \tilde{C}_s & & \\ 0 & & & & \\ 0 & & & & \end{array}\right]. \tag{2.16}$$

Therefore each of the three factors of $\hat{L}$ will have the same structure and, in particular

$$\hat{L} = \left[\begin{array}{c|cccc} \left(1+2i\alpha \frac{\Delta t}{\Delta x} u\xi\sqrt{1-\xi^2}\right)\left(1+2i\alpha \frac{\Delta t}{\Delta y} v\eta\sqrt{1-\eta^2}\right)\left(1+2i\alpha \frac{\Delta t}{\Delta z} w\zeta\sqrt{1-\zeta^2}\right) & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & & \tilde{\hat{L}} & \\ 0 & & & & \end{array}\right] \tag{2.17}$$

where $\xi = \sin(\omega_1/2)$, $\eta = \sin(\omega_2/2)$, $\zeta = \sin(\omega_3/2)$ and $\omega_1, \omega_2, \omega_3$ are the dual Fourier variables ranging from zero to 2π and hence $-1 \leq \xi, \eta, \zeta \leq 1$. Clearly, $\hat{R}$ will also have the same type structure - specifically

$$\hat{R} = \left[\begin{array}{c|cccc} 2i \frac{\Delta t}{\Delta x}\left(u\xi\sqrt{1-\xi^2} + \frac{\Delta x}{\Delta y} v\eta\sqrt{1-\eta^2} + \frac{\Delta x}{\Delta z} w\zeta\sqrt{1-\zeta^2}\right) & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & & \tilde{\hat{R}} & \\ 0 & & & & \end{array}\right] \tag{2.18}$$

It follows that $\hat{L}^{-1}\hat{R}$ and $\hat{R}\hat{L}^{-1}$ will also have this type of structure although their lower right blocks will differ, $(\tilde{\hat{R}\hat{L}^{-1}}) \neq (\tilde{\hat{L}^{-1}\hat{R}})$. Thus, finally, we conclude that while $G_\Delta \neq G_Q$ they do share the indicated matrix structure with $G_{\Delta 11} = G_{Q11}$. In particular

$$G_\Delta = \left[\begin{array}{c|cccc} g_{11} & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & I + (\tilde{\hat{L}^{-1}\hat{R}}) & & \\ 0 & & & & \end{array}\right] . \tag{2.19}$$

where

$$g_{11} = 1 + \frac{2i(\Delta t/\Delta x)\left[u\xi\sqrt{1-\xi^2} + (\Delta x/\Delta y)v\eta\sqrt{1-\eta^2} + (\Delta x/\Delta z)w\zeta\sqrt{1-\xi^2}\right]}{\left[1+2i\alpha(\Delta t/\Delta x)u\xi\sqrt{1-\xi^2}\right]\left[1+2i\alpha(\Delta t/\Delta y)v\eta\sqrt{1-\eta^2}\right]\left[1+2i\alpha(\Delta t/\Delta z)w\zeta\sqrt{1-\zeta^2}\right]} .$$

The corresponding lower right block of G_Q will be $I + (\tilde{\hat{R}\hat{L}^{-1}})$.

We are now ready to establish the unconditional instability of G_Δ (and, hence, also of G_Q). G_Δ may be written as

$$G_\Delta = \left[\begin{array}{c|cccc} g_{11} & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & \tilde{g} & & \\ 0 & & & & \end{array}\right] , \tag{2.20}$$

where the definitions of g_{11} and $\tilde{g}$ are clear from (2.19). The instability of scheme will be established if we show that $\|G_\Delta\| > 1$, with the square of the norm given by

$$\|G_\Delta\|^2 = \sigma\left(G_\Delta^T G_\Delta\right), \tag{2.21}$$

$\sigma(\cdot)$ being the spectral radius of the designated matrix. In our case

$$G_\Delta^T G_\Delta = \left[\begin{array}{c|ccc} g_{11} & 0 & 0 & 0 \\ \hline 0 & & & \\ 0 & & \tilde{g}^T & \\ 0 & & & \end{array}\right] \cdot \left[\begin{array}{c|ccc} g_{11} & 0 & 0 & 0 \\ \hline 0 & & & \\ 0 & & \tilde{g} & \\ 0 & & & \end{array}\right] = \left[\begin{array}{c|ccc} g_{11} & 0 & 0 & 0 \\ \hline 0 & & & \\ 0 & & \tilde{g}^T\tilde{g} & \\ 0 & & & \end{array}\right],$$

and thus

$$\sigma(G_\Delta^T G_\Delta) = \max\left(g_{11}^2;\ \|\tilde{g}^T\tilde{g}\|\right). \tag{2.22}$$

If we can demonstrate that for any $0 < \Delta t$ there is a triplet ξ, η, ζ such that $g_{11} > 1$ then instability has been established. In other words, we have just shown that if a scalar version of (2.10) and (2.11) is unstable then also the delta-form of the Backward Euler scheme is unstable for the equations of gas-dynamics (Euler's equations). The same argument, of course, holds true also for the LODQ algorithm. The scalar version of (2.10) which gives rise to an amplification factor $g = g_{11}$ is

$$\frac{\partial U}{\partial t} + u\,\frac{\partial U}{\partial x} + v\,\frac{\partial U}{\partial y} + w\,\frac{\partial U}{\partial z} = 0, \tag{2.23}$$

where now U is the unknown scalar and u,v,w are scalar coefficients, which for the purpose of (linear) stability analysis may be taken as (locally) constant. We can now "stretch" the coordinates x,y,z through division respectively by u, $(\Delta x/\Delta y)v$, and $(\Delta x/\Delta z)w$ so that the corresponding amplification factor has the form (with $\lambda = \Delta t/\Delta x$)

$$g = g_{11} = 1 + \frac{2i\lambda\left(\xi\sqrt{1-\xi^2} + \eta\sqrt{1-\eta^2} + \zeta\sqrt{1-\zeta^2}\right)}{\left(1+2i\alpha\lambda\xi\sqrt{1-\xi^2}\right)\left(1+2i\alpha\lambda\eta\sqrt{1-\eta^2}\right)\left(1+2i\alpha\lambda\zeta\sqrt{1-\zeta^2}\right)}, \tag{2.24}$$

define

$$a = 2\lambda\left(\xi\sqrt{1-\xi^2} + \eta\sqrt{1-\eta^2} + \zeta\sqrt{1-\zeta^2}\right)$$

$$b = 4\alpha^2\lambda^2\left(\xi\eta\sqrt{1-\xi^2}\sqrt{1-\eta^2} + \eta\xi\sqrt{1-\eta^2}\sqrt{1-\zeta^2} + \zeta\xi\sqrt{1-\zeta^2}\sqrt{1-\xi^2}\right)$$

$$e = 8\alpha^3\lambda^3\xi\eta\zeta\sqrt{1-\xi^2}\sqrt{1-\eta^2}\sqrt{1-\zeta^2}$$

and thus rewrite (2.24) as:

$$g = 1 + \frac{ia}{1 + i\alpha a - b - ie} = \frac{(1-b) - i\left(e-(\alpha+1)a\right)}{(1-b) - i(e-\alpha a)} \tag{2.25}$$

and

$$|g|^2 = \frac{(1-b)^2 + e^2 - 2(1+\alpha)ea + (1+\alpha)^2a^2}{(1-b)^2 + e^2 - 2\alpha ae + \alpha^2a^2} \,. \tag{2.26}$$

For instability we require to show $|g|^2 > 1$, or

$$-2(1+\alpha)ea + (1+2\alpha+\alpha^2)a^2 > -2\alpha ea + \alpha^2a^2,$$

or

$$a[-2e + (1+2\alpha)a] > 0. \tag{2.27}$$

This inequality is satisfied by an infinite number of triplets (ξ,η,ζ). For example, take $0 < \xi < 1\left(0<\omega_1<2\pi\right)$, $\eta = -\zeta\left(\omega_2 = -\omega_3\right)$, then $a = 2\left(\lambda\xi\sqrt{1-\xi^2}\right) > 0$ and $e = -8\alpha^3\lambda^3\xi\eta^2\sqrt{1-\xi^2}(1-\eta^2) < 0$; the inequality is then satisfied for all $-\frac{1}{2} < \alpha$. (In real computation, of course, $\alpha > 0$.) This completes the proof that G_Δ, the amplification matrix for the delta-form of the three-dimensional approximate-factored Backward Euler scheme for the gas-dynamic equations has a norm exceeding unity and hence is unstable. As shown above, the argument holds also for G_Q.

The question now is can one modify the algorithms (2.12) and (2.14) in such a way that, at least, $g_{11} \leq 1$? Success will not necessarily establish stability for the complete set of the equations of gas-dynamics (Euler equations) but at least if the scalar case is shown to be stabilized there is hope that this will hold true also for Euler's equations. We will carry out the manipulations in the Fourier space and when $g = g_{11}$ is stabilized we will inquire as to how these changes map into the physical space in which the algorithms (2.12) and (2.14) are written. Examination of (2.24) convinced us to change g in such a way that the modified amplification factor, g_m will have the form

$$g_m = 1 - \kappa(\xi^4+\eta^4+\zeta^4) + \frac{2i\lambda\left(\xi\sqrt{1-\xi^2} + \eta\sqrt{1-\eta^2} + \zeta\sqrt{1-\zeta^2}\right)(\xi^2+\eta^2+\zeta^2)}{\left(1+2i\lambda\alpha\xi\sqrt{1-\xi^2}+\beta\xi^2\right)\left(1+2i\lambda\alpha\eta\sqrt{1-\eta^2}+\beta\eta^2\right)\left(1+2i\lambda\alpha\zeta\sqrt{1-\zeta^2}+\beta\zeta^2\right)}. \tag{2.28}$$

Numerical tests were conducted for $\alpha = 1/2$ and 1 and for $1/2 \leq \beta \leq 5$ with $-1 \leq \xi,\eta,\zeta \leq 1$ taken over this range by steps of $\Delta\xi = \Delta\eta = \Delta\zeta = 0.05$; and various values of λ ranging from 1 to 10^4. Uniform stability ($g_m \leq 1$ for all ξ,η,ζ and λ) was found for κ ranging between 0.05 and 0.5. We do not have analytical proofs for these bounds but the results of the numerical experimentation leave little doubt as to their validity. (Note that for $\lambda \gg 1$ and upper bound on κ is 2/3.)

Next we'll consider what changes to introduce into the nonlinear nonsymmetrized version of (2.14) in order that the first variation of the new algorithm will have (in the scalar case) the amplification factor g_m given by (2.28).

Taking note of the fact that ξ^2 and ξ^4 in Fourier space correspond, respectively, to $-\Delta x^2\delta_x^2/4$ and $\Delta x^4\delta_x^4/16$ and similarly for $\eta^2,\eta^4,\zeta^2,\zeta^4$, we modify (2.14) as follows:

$$L_m \vec{U}^{n+1} = \vec{Q}^n \tag{2.29a}$$

$$\vec{Q}^{n+1} = \left[1 - \frac{\kappa\Delta x^4}{16}\left(\delta_x^4 + \left(\frac{\Delta y}{\Delta x}\right)^4\delta_y^4 + \left(\frac{\Delta z}{\Delta x}\right)^4\right)\delta_z^4\right]\vec{Q}^n + R\vec{V}^{n+1}, \tag{2.29b}$$

where

$$L_m = \left(I + \alpha\Delta t A\delta_x\mu_x - \beta(\Delta x^2/4)\delta_x^2\right)\left(I + \alpha\Delta t B\delta_y\mu_y - \beta(\Delta y^2/4)\delta_y{}^2\right)$$

$$\left(I + \alpha\Delta t C\delta_z\mu_z - \beta(\Delta z^2/4)\delta_z^2\right), \tag{2.30}$$

and

$$\vec{V}^{n+1} = -\frac{\Delta x^2}{4}\left[\delta_x^2 + \left(\frac{\Delta y}{\Delta x}\right)^2\delta_y^2 + \left(\frac{\Delta z}{\Delta x}\right)^2\delta_z^2\right]\vec{U}^{n+1}. \tag{2.31}$$

Notice that at steady state the solution vector is the newly defined quantity $\vec{V}$ and not $\vec{U}$, unlike the case in the unmodified (and unstable) formulation. The amplification matrix of (2.29) is

$$G_{Qm} = \left[1 - \kappa(\xi^4+\eta^4+\zeta^4)\right]I + \hat{R}\hat{L}_m^{-1}(\xi^2+\eta^2+\zeta^2), \tag{2.32}$$

and for the scalar case this reduces to g_m given by (2.28) which numerical experiments have shown to be stable.

In order to stabilize the three-dimensional "delta"-form, equation (2.12) is modified as follows:

$$L_m(\vec{U}^{n+1}-\vec{U}^n) = -\frac{\kappa\Delta x^4}{16} L_m\left[\delta_x^4 + \left(\frac{\Delta y}{\Delta x}\right)^4\delta_y^4 + \left(\frac{\Delta z}{\Delta x}\right)^4\delta_z^4\right]\vec{U}^n + R\vec{V}\ , \qquad (2.33)$$

to give the amplification matrix

$$G_{\Delta m} = \left[1 - \kappa(\xi^4+\eta^4+\zeta^4)\right]I + \hat{L}_m^{-1}\hat{R}(\xi^2+\eta^2+\zeta^2). \qquad (2.34)$$

At steady state (if one is achieved) the vector $\vec{V}$ solves

$$R\vec{V} = \frac{\kappa\Delta x^4}{16\Delta t} L_m\left[\delta_x^4 + \left(\frac{\Delta y}{\Delta x}\right)^4\delta_y^4 + \left(\frac{\Delta z}{\Delta x}\right)^4\delta_z^4\right]\vec{U}^n, \qquad (2.35)$$

rather than $R\vec{V} = 0$. The question arises as to the error introduced into the solution (2.35) by its right-hand side. For the solution of (2.35) to be second-order accurate it is necessary for the right-hand side to be of $O(\Delta x^2)$. For $\vec{U}$ sufficiently smooth (e.g., $\nabla^4\vec{U} \approx O(1)$) the error is determined, for large Δt, by the term of L_m of highest power in Δt, multiplied by the coefficient in front of it. Similarly one could get an estimate for small Δt. This argument then predicts that (2.35) will yield steady-state solutions which are second-order accurate spatially for the following range of λ:

$$\frac{\kappa}{16N} < \lambda < \frac{4N^{1/2}}{\kappa^{1/2}\alpha^{3/2}}\ , \qquad (2.36)$$

where N is $1/\Delta x$. For nonuniform grid this estimate will change somewhat, but not materially. This estimate yields a wide range of available λ for second-order accuracy. Numerical evidence in Section 4 supports the estimate (2.36) and indicates that these limits exceed the practical range.

3. PROGRAMMING CONSIDERATIONS

Before embarking on a discussion of numerical results obtained with the scheme proposed in Section 2, we will first discuss some aspects of implementing the scheme. In the numerical results to be presented in Section 4, the algorithm defined in Eqs. (2.29) - (2.31) was used, but in a slightly modified form. The first modification concerned the quantity $\vec{V}$ defined by Eq. (2.31). In the computer program $\vec{V}$ was defined as

$$\vec{V}^n = \mu\vec{U}^n - (1-\mu)\frac{\Delta x^2}{4}\left[\delta_x^2 + \left(\frac{\Delta y}{\Delta x}\right)^2\delta_y^2 + \left(\frac{\Delta z^2}{\Delta x^2}\right)\delta_z^2\right]\vec{U}^n, \tag{3.1}$$

a modification which allowed us to examine the effect of the discrete Laplacian operating on $\vec{U}$ both on the accuracy and stability of the scheme. Again, if steady state is achieved, it is the quantity $\vec{V}^n$ defined as above which satisfies the steady difference equations in the sense of Eq. (2.35). The numerical tests described in connection with Eq. (2.28) were extended to include $0 \leq \mu \leq 1$. All stable runs remained stable except for $\mu = 1$ which destabilized the scheme.

An additional difficulty that had to be overcome in the implementation of the stabilized LODQ algorithm was the reintroduction of the Yanenko boundary condition errors [4], [6] through the terms proportional to κ in Eq. (2.29b). This difficulty can be circumvented by rewriting (2.29b) in the form

$$\vec{Q}^{n+1} = \vec{Q}^n - \frac{\kappa\Delta x^4}{16}\left[\delta_x^4 + \left(\frac{\Delta y}{\Delta x}\right)^4\delta_y^4 + \left(\frac{\Delta z}{\Delta x}\right)^4\delta_z^4\right]L_m\vec{U}^{n+1} + R\vec{V}^{n+1}. \tag{3.2}$$

Use of the corrector step of the algorithm in this form, although formally identical to (2.29b) on an infinite domain, prevents introduction of $0(1)$ contributions to the residual on grid lines near computational domain boundaries (see [4]).

The next point warranting discussion in this section concerns the implementation of the algorithm near boundaries. On grid lines immediately adjacent to boundaries one cannot directly apply Eq. (3.2) as this would require data from outside the domain (since the fourth difference operators as defined herein require a five point stencil). On grid lines immediately adjacent to domain boundaries, a second-order accurate six point uncentered difference operator is used in lieu of the centered five point operator as required. In addition, $L_m \vec{U}^{n+1}$, which is evaluated explicitly as

$$L_m \vec{U}^{n+1} = L_1[L_2(L_3 \vec{U}^{n+1})] \tag{3.3}$$

is required at all points on the grid including boundary points in order to operate on this quantity with the discrete biharmonic. We consider a computational domain discretized uniformly in Cartesian space with $J \cdot K \cdot L$ grid points such that

$$\begin{aligned} &1 \leq j \leq J \\ &1 \leq k \leq K \\ &1 \leq \ell \leq L \ . \end{aligned} \tag{3.4}$$

The evaluation of (3.3) is accomplished in the calculations presented herein as follows:

(i) Evaluate $L_3 \vec{U}^{n+1}$ over the range $1 \leq j \leq J$, $1 \leq k \leq K$ and $2 \leq \ell \leq L-1$ using central differences. Evaluate $L_3 \vec{U}^{n+1}$ for $1 \leq j \leq J$, $1 \leq k \leq K$ and $\ell = 1, L$ using second-order accurate forward or backward differences as required.

(ii) Evaluate $L_2(L_3 \vec{U}^{n+1})$ for $1 \leq j \leq J$, $2 \leq k \leq K-1$ and $1 \leq \ell \leq L$.

(iii) Evaluate $L_1[L_2(L_3 \vec{U}^{n+1})]$ for $2 \leq j \leq J-1$, $2 \leq k \leq K-1$, $1 \leq \ell \leq L$.

(iv) Use second-order accurate extrapolation to evaluate $L_m \vec{U}^{n+1}$ from interior values for $j=1,J$, $k=1,K$, $\ell=1,L$.

Note that the added complexity of calculating $L_m \vec{U}^{n+1}$ is required only by the new algorithm in the LODQ form which was the case considered here. For the new algorithm in the delta-form, an evaluation of $L_m \nabla^4 \vec{U}$ is required but details of the evaluation of this term were not studied.

The final point to be considered is the application of boundary conditions. Since it is the vector $\vec{V}$ which satisfies the steady difference equations over the domain, the appropriate boundary conditions for $\vec{V}$ are the boundary conditions derived from the steady continuum problem. For a first order system of the type considered here modeled with central differences, two types of boundary conditions can arise, here termed physical and numerical. Physical boundary conditions arise from application of the continuum boundary conditions to the discrete problem while numerical boundary conditions arise from the need to close the system of difference equations on computational domain boundaries not corresponding to physical boundaries. In the calculations presented in Section 4 the physical boundary conditions were always taken to be a specification of $\vec{V}$ over some segment of the boundary of the computational domain. The numerical boundary conditions were always taken to be a second-order implicit extrapolation of $\vec{V}$.

In addition to boundary conditions on $\vec{V}$, the new algorithm requires boundary data on $\vec{U}$ in either LODQ or Δ forms. In the present work these conditions on $\vec{U}$ were taken from the conditions on $\vec{V}$. Where physical boundary conditions were applied the condition on $\vec{U}$ was taken to be the same as the condition on $\vec{V}$. Application of numerical boundary conditions was, however, somewhat more complicated. Let us consider the plane $j = J$ to be

an artificial computational domain boundary at which we apply

$$\vec{V}^n_{J,k,\ell} = 2\vec{V}^n_{J-1,k,\ell} - \vec{V}^n_{J-2,k,\ell}. \tag{3.5}$$

If we were to directly substitute into Eq. (3.5) the definition of $\vec{V}$ in terms of $\vec{U}$, Eq. (3.1), a formula involving $\vec{U}^n_{J+1,k,\ell}$ would arise. Thus, a boundary condition for $U^n_{J,k,\ell}$ cannot be derived solely from the boundary condition on $\vec{V}^n_{J,k,\ell}$, but instead, an additional numerical boundary condition on $\vec{U}^n$ must be specified. This result is not surprising since Eq. (2.29b) written entirely in terms of $\vec{V}^n$ and $\vec{Q}^n$ spreads the stencil of $\vec{V}^n$ to five points in each direction, indicating the need for an additional numerical boundary condition on $\vec{U}^n$. This extra condition was here taken to be

$$\vec{U}^n_{J+1,k,\ell} = 2\vec{U}^n_{J,k,\ell} - \vec{U}^n_{J-1,k,\ell}. \tag{3.6}$$

Substitution of Eqs. (3.1) and (3.6) into Eq. (3.5), expanding the definitions of the operators and collecting terms yields the following condition for $\vec{U}^n_{J,k,\ell}$;

$$\begin{aligned}\left[\mu + \frac{3}{2}(1-\mu)\right]\vec{U}^n_{J,k,\ell} &= \left[2\mu + \frac{13}{4}(1-\mu)\right]\vec{U}^n_{J-1,k,\ell} - \left[\mu+2(1-\mu)\right]\vec{U}^n_{J-2,k,\ell} + \frac{1}{4}(1-\mu)\vec{U}^n_{J-3,k,\ell} \\ &\quad + \frac{1}{4}(1-\mu)\Delta x^2\delta_x^2\left(\vec{U}^n_{J-1,k+1,\ell} + \vec{U}^n_{J-1,k-1,\ell} + \vec{U}^n_{J-1,k,\ell+1} + \vec{U}^n_{J-1,k,\ell-1}\right).\end{aligned} \tag{3.7}$$

Equation (3.7) represents the boundary condition on $\vec{U}^n_{J,k,\ell}$ consistent with

both Eqs. (3.5) and (3.6). Implementation of this boundary condition in terms of the algorithmic interpretation of Eq. (2.29a) would couple adjacent tridiagonal inversions. This undesirable property can be eliminated by noting that the quantity in brackets in the last term on the right hand side of (3.7) can be replaced by $4\vec{U}^{n}_{J-1,k,\ell}$ to $0(\Delta y^2,\Delta z^2)$. Thus, the final form of the boundary condition used in the present work is

$$\left[\mu + \frac{1}{2}(1-\mu)\right]\vec{U}^{n}_{J,k,\ell} = \left[2\mu + \frac{5}{4}(1-\mu)\right]\vec{U}^{n}_{J-1,k,\ell} - \vec{U}^{n}_{J-2,k,\ell} + \frac{1}{4}(1-\mu)\vec{U}^{n}_{J-3,k,\ell}. \tag{3.8}$$

4. NUMERICAL RESULTS

4.1. Test Problem

The test problem examined in the present work involved the solution of Eq. (2.23) on the cube $0 \leq x,y,z \leq 2\pi$ with $u = v = w = 1$ and subject to steady boundary data on three faces of the cube. These data and their location were taken to be

$$U = \cos(-y) + \cos(y-z) + \cos(-z) \qquad \begin{cases} x = 0 \\ 0 \leq y,\ z \leq 2\pi \end{cases}$$

$$U = \cos(x) + \cos(-z) + \cos(x-z) \qquad \begin{cases} y = 0 \\ 0 \leq x,\ z \leq 2\pi \end{cases}$$

$$U = \cos(x-y) + \cos(y) + \cos(x) \qquad \begin{cases} z = 0 \\ 0 \leq x,\ y \leq 2\pi. \end{cases}$$

The exact steady solution to this problem is given by

$$u = \cos(x-y) + \cos(y-z) + \cos(x-z).$$

Numerical solutions to the steady problem were obtained by solving Eq. (2.23) in time until an asymptotic steady state was reached. The initial data was taken to be

$$U = \cos(x-y) + \cos(y-z) + \cos(x-z) + xyz(x-2\pi)(y-2\pi)(z-2\pi) ,$$

$$0 \leq x,y,z \leq 2\pi .$$

The steady state was assumed to be reached when L_2 norm of the residual (i.e., RV^n) was reduced below a set tolerance. The accuracy of the resultant solution was measured by calculating the maximum error and the L_2 norm of the error between the exact and discrete solutions.

4.2. Discussion of Results

In order to assess the accuracy of the three-dimensional algorithm given by Eq. (2.29) - (2.31) a series of runs were made for the test problem for a sequence of grids for $\kappa = 0.1$, a value sufficient to insure stability. These results are summarized in Figure 1 where second-order accuracy is seen to be obtained.

The effect of λ on the convergence rate of the scheme is shown in Figure 2. Note the strong dependence of the convergence rate on λ and the fact that the optimum λ for convergence is $\lambda \approx 1$. Similar results have been reported elsewhere.

Finally, in Figure 3, the effect of λ on the solution accuracy for fixed $\kappa \neq 0$ is shown. The accuracy as defined in the last paragraph of Section 4.1 is also compared with the accuracy of the $\kappa = 0$ case. Note that

although as a pure initial value solver the algorithm is unstable for $\kappa = 0$, for the complete range of λ, it appears that this case can be stabilized by the boundary conditions. Numerical solutions obtained in the range $\lambda \approx 0(1)$ with $\kappa = 0$ are thus taken to be solutions of the steady difference operator, $RV^n = 0$. Note the fact that there is a minimum in error produced by the scheme (2.29) - (2.31) which is lower than that produced by the scheme with $\kappa = 0$. This fact can be explained by noting that the temporally inconsistent terms in the algorithm have coefficients λ^{-1}, 1, λ, λ^2. Hence at small values of λ, the λ^{-1} terms will dominate the error while the λ^2 terms will come to dominate at larger values of λ. Thus, the error should have a minimum value at some value of λ. It should also be noted that both the analysis and computational evidence verify that the parameters μ and β have no effect on the steady solution.

A final point should be made about stability of the algorithm with $\kappa = 0$, $\mu = 1$ and $\beta = 0$ (i.e., the basic algorithm). The computational evidence obtained in the present investigation indicates that the instability of this interior point scheme is quite mild. On finite grids it is possible to obtain solutions for small λ, although the maximum stable value of λ appears to decrease as the gird is refined. As an example, on a grid of $33\times33\times33$ for the test problem an effective spectral radius of 1.002 (defined as $\left(\Delta Q^n_{max}/\Delta Q^1_{max}\right)^{1/n}$) was found at $\lambda = 100$. By contrast, the new algorithm was found to have a spectral radius below unity for $1 \leq \lambda < 10^6$. The basic scheme is already extremely weakly unstable at $\lambda = 5$, however, and in effect will never converge or diverge, i.e., the effective spectral radius of the scheme is unity. Since the instability of the interior point scheme is so mild, the implicit damping introduced by the parameter β is very effective in accelerating the convergence of the scheme when it is stable, but the

parameter κ, which introduces a form of explicit damping that does not vanish in the steady state, is most effective in extending the stable range of λ.

5. SUMMARY AND DISCUSSION

The following main reuslts were obtained in this paper:

1. It has been shown that both the LODQ (2.14) and the Beam-Warming (2.11) algorithms for the Euler equations of gas-dynamics in three-dimensions are unconditionally (linearly) unstable.

2. A stable algorithm for three-dimension is proposed. It does not have any more implicit operations than either the LODQ or Beam-Warming schemes.

3. This stable algorithm is second-order accurate for a wide range of the Courant number.

4. It was observed that all three algorithms (LODQ, Beam-Warming and the new one) have a convergence rate to steady state which is not monotonic with λ (unlike an unfactored implicit algorithm). We were surprised to find that the optimal Courant number λ for which convergence to steady state is fastest is of order unity for both two-dimensional and three-dimensional cases.

The observation mentioned in the last point above raises the issue of the efficacy of implicit factored schemes verses explicit ones. Clearly for problems with uniform mesh it would seem that the operation count and CPU time favor explicit algorithms. It is not yet clear which way to go in the case of highly stretched mesh, especially if time accuracy is not important.

All of the above evidence and analysis for the new algorithm were for the linear scalar wave equation. Clearly much experimentation is necessary to explore its application to the Euler equations in three-dimensions with uniform and non-uniform grids.

REFERENCES

[1] ABARBANEL, S. and GOTTLIEB, D. 1981 Optimal time splitting for two- and three-dimensional Navier-Stokes equations with mixed derivatives. J. Comput. Phys., **41**, 1-33. (Also ICASE Report No. 80-6, February 1980).

[2] BEAM, R. M. and WARMING, R. F. 1976 An implicit finite difference algorithm for hyperbolic systems in conservation law form. J. Comput. Phys. **22**, September, 87-110.

[3] DOUGLAS, J. and GUNN, J. E. 1964 A general formulation of alternating direction methods. Numer. Math. **6**, 428-453.

[4] DWOYER, D. L. and THAMES, F. C. 1981 Temporal and spectral inconsistencies of time-split finite-difference schemes. NASA TP-1790.

[5] WARMING, R. F. and BEAM, R. M. 1979 Extension of a stability to ADI methods. BIT **19**, 395-417.

[6] YANENKO, N. N. 1971 The Method of Fractional Steps. (Northrop Corporate Labs, transl.) Springer-Verlag.

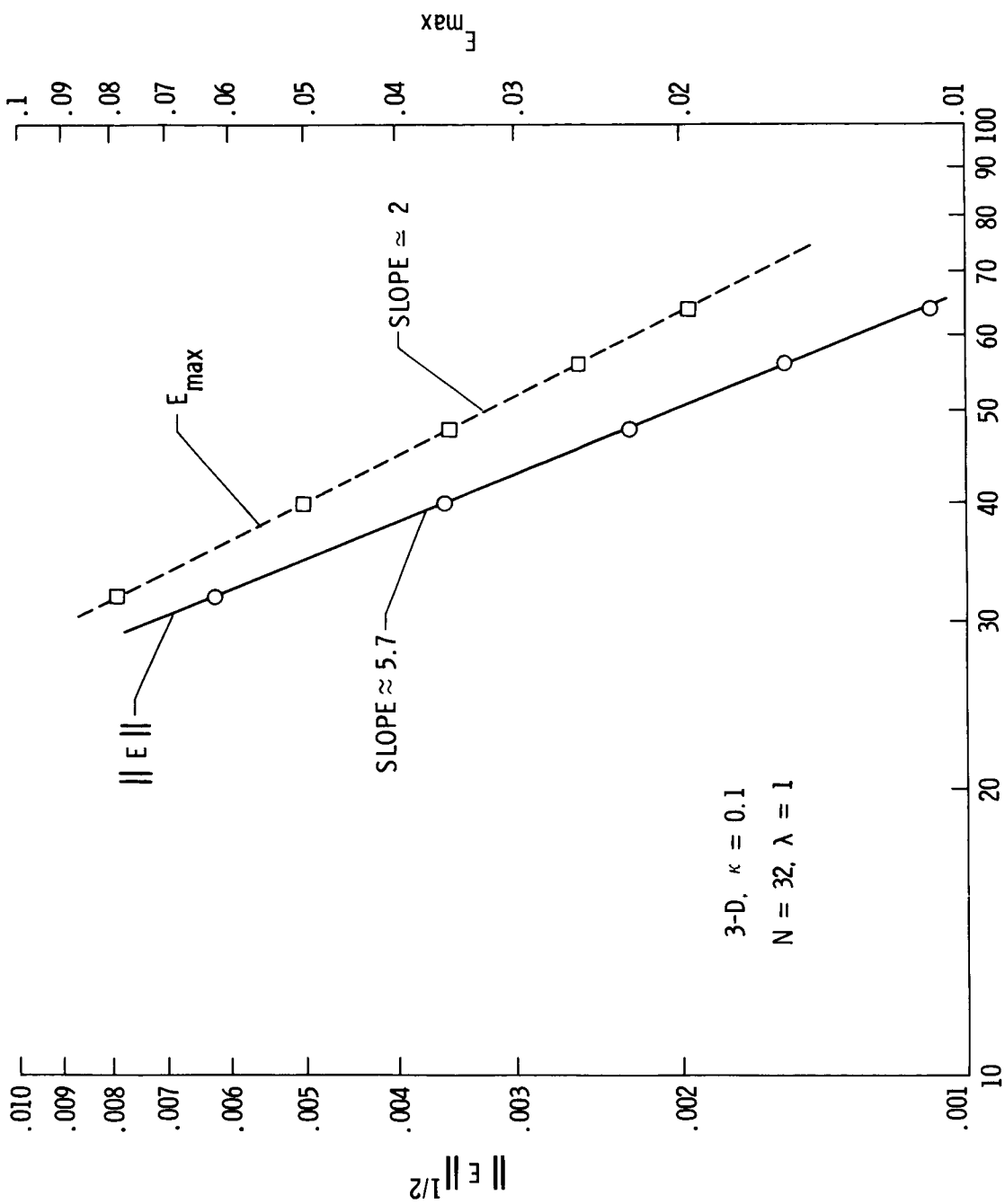

Figure 1. Spatial convergence rate of stable implicit scheme for three-dimensional wave equation.

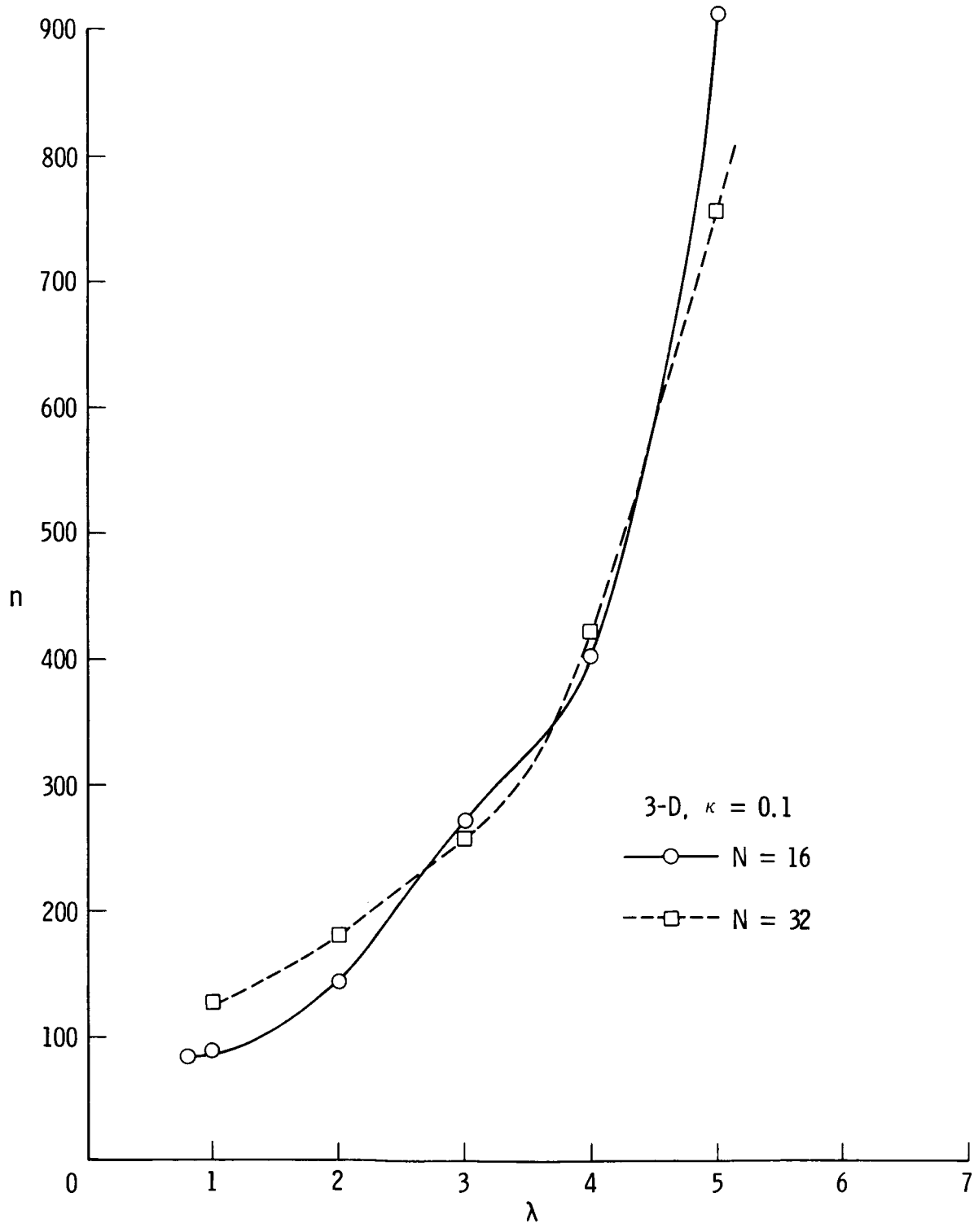

Figure 2. Effect of Courant number on iterative convergence of stable implicit scheme for three-dimensional wave equation.

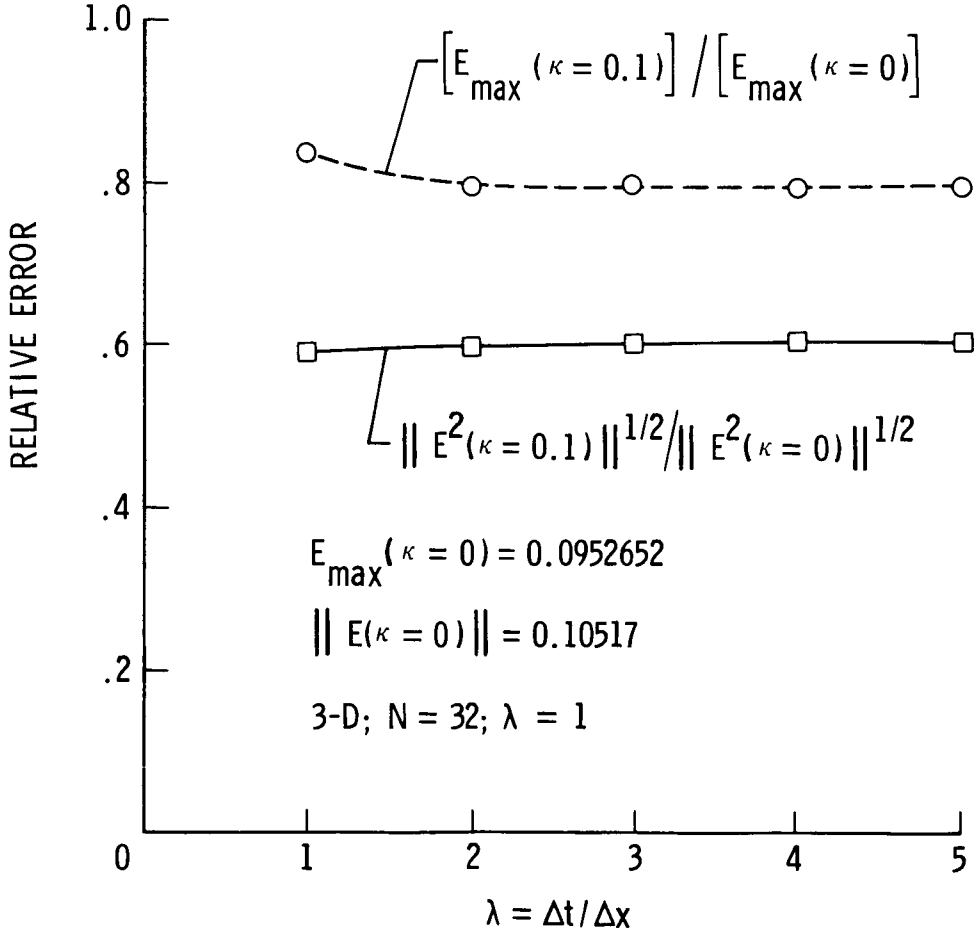

Figure 3. Effect of Courant number on spatial truncation error of stable implicit scheme for three-dimensional wave equation.